1   When performed, each operation 1014 or 1016 enters operation 1018, which sets IFAR to

2   the "Execution IFAR" value, and the execution results obtained for all instructions

3   following the branch are flushed from instruction pipeline.

4   Then, operation 1019 sets a BHT_write_addr register 318 to the "address of the branch"

5   field obtained from the BIQ entry for the executed branch.   The BHT_write_data is set to

6   1, if the branch outcome is "taken", else it is set to 0, and this value is written over the

7   current BHT bit in the BHT to insure that it is corrected.

8   The next operation 1021 is then performed, and it releases the BIQ entry for the executed

9   branch instruction by setting its valid bit to the "0" state. The process then goes to

10   FIGURE 8 entry point A to repeat its operation which has been previously described

11   herein..

12   While I have described the preferred embodiment of my invention, it will be understood

13   that those skilled in the art, both now and in the future, may make various improvements

14   and enhancements which fall within the scope of the claims, which follow. These claims

15   should be construed to maintain the proper protection for the invention first disclosed

16   here.

17   The invention claimed is:

1   1. A branch prediction feature in a computer system for improving branch prediction rate

2   when utilizing a branch history table (BHT) with an instruction cache, and being able to

3   use branch history developed for instruction lines replaced in the instruction cache,

4   comprising

5   the instruction cache (I-cache) comprised of I-cache rows for respectively storing

6   instruction lines fetched from a second level cache (L2 cache) in a storage hierarchy of the

7   computer system,

8   the L2 cache comprised of L2 rows for respectively storing instruction lines fetched from a

9   system main storage in the storage hierarchy of the computer system,

10   a hint instruction location being provided in each I-cache row, and a hint instruction

11   location being provided in each L2-cache row,

12   in response to an I-cache hit for a current instruction in an I-cache line, a hint processor

13   for generating a hint instruction associated with the I-cache line, the hint instruction

14   containing: a BHT entry and a branch mask indicating the location(s) of each branch

15   instruction in the associated I-cache line, and a current index of the I-cache line in the

16   I-cache, the hint processor storing the hint instruction in the hint instruction location in the

17   current I-cache row for the current instruction,

18   in response to an I-cache miss, an L2 row containing a copy of the current I-cache

19   instruction line and an associated hint instruction being located in the L2 cache and copied

20   into the current I-cache row, and forwarding the hint instruction to the hint processor to

21   be executed, and

1   a current BHT entry containing a prediction bit for the current instruction when the

2   current instruction is a branch instruction, and the prediction bit being changed if the

3   current instruction is determined to be incorrectly predicted for the current instruction.

4   2. A branch prediction feature for improving the branch prediction rate in a computer

5   system as defined in claim 1, further comprising

6   the I-cache being located in a chip containing a central processor of the computer system,

7   and the current I-cache row being located at an I-cache index determined by a current

8   instruction address in an instruction fetch address register (IFAR) for the current

9   instruction being selected for execution in the chip.

10   3. A branch prediction feature for improving the branch prediction rate in a computer

11   system as defined in claim 2, further comprising

12   the I-cache being formed of two subarrays, one subarray containing a row for each

13   instruction line in the I-cache, and the second subarray containing a row for each hint

14   instruction in the I-cache, the I-cache index being used simultaneously in both subarrays to

15   locate the current instruction line in one subarray and an associated hint instruction in the

16   other subarray.

1   4. A branch prediction process for a computer system for improving branch prediction rate

2   when using a branch history table, comprising

3   determining if a program instruction processor (processor) has an access hit (hit) or access

4   miss (miss) in an instruction cache (I-cache) when utilizing an instruction address (IFAR

5   address) in attempting to select a program instruction (instruction) for execution by the

6   processor,

7   generating a hint instruction (when the instruction is a branch) in response to a hit

8   occurring during the determining operation, storing the hint instruction in association with

9   a copy of an instruction line containing the instruction in a storage hierarchy of the

10   computer system, the hint instruction storing BHT prediction fields obtained from a copy

11   of a current BHT entry associated with the instruction line when the hit occurs, and storing

12   a branch mask in the hint instruction for locating an associated BHT field (indicating the

13   BHT field associated with the location of the instruction in the instruction line), and

14   transferring the copy of the instruction line and associated hint instruction from the

15   storage hierarchy to the I-cache in response to a miss occurring during the determining

16   operation, and executing the hint instruction to restore a BHT prediction field in a current

17   BHT entry to the state of a BHT field in the hint instruction located by the branch mask.

18   5. A branch prediction process for a computer system using a branch history table as

19   defined by claim 4, further comprising

1   also generating a bht index field in the hint instruction by storing an I-cache index (current

2   I-cache index) that locates the instruction line in the I-cache and also locates the BHT entry

3   (current BHT entry) associated with the instruction line.

4   6. A branch prediction process for a computer system using a branch history table as

5   defined by claim 5, further comprising

6   initially providing a hint instruction no-operation code (NOP code) in the hint instruction

7   to allocate space for a hint instruction in association with an instruction line.

8   7. A branch prediction process for a computer system using a branch history table as

9   defined by claim 6, further comprising

10   providing a hint instruction operation code (op code) in the hint instruction to identify it as

11   a hint instruction when fields of the hint instruction are provided.

12   8. A branch prediction process for a computer system for improving branch prediction rate

13   when using a branch history table as defined by claim 4, further comprising

14   locating the copy of the instruction line in the storage hierarchy by: detecting a line address

15   field in an I-cache directory entry associated with the instruction line in the I-cache,

16   composing an address for the copy of the instruction line in the storage hierarchy by

17   combining the line address field obtained by the detecting operation and an I-cache index

18   obtained from the IFAR address of the instruction.

1   9. A branch prediction process for a computer system for improving branch prediction rate

2   when using a branch history table as defined by claim 8, further comprising

3   accessing the copy of the instruction line and the associated hint instruction in a second

4   level cache (L2 cache) in the storage hierarchy.

5   10. A branch prediction process for a computer system for improving branch prediction

6   rate when using a branch history table as defined by claim 9, further comprising

7   storing in a main memory of the computer system a copy of the instruction line and the

8   associated hint instruction when a L2 cache miss occurs to the storage line requiring a

9   replacement of the storage line and the associated hint instruction in the L2 cache.

10   11. A branch prediction process for a computer system for improving branch prediction

11   rate when using a branch history table as defined by claim 10, further comprising

12   retrieving from the main memory both the copy of the instruction line and the associated

13   hint instruction when a L2 cache miss occurs to the storage line requiring a retrieval of the

14   storage line from the storage hierarchy.

15   12. A branch prediction process for a computer system for improving branch prediction

16   rate when using a branch history table as defined by claim 9, further comprising

17   storing in a main memory of the computer system a copy of the instruction line without a

18   copy of the associated hint instruction when a L2 cache miss occurs to the storage line

19   requiring a replacement of the storage line and the associated hint instruction in the L2

20   cache to lose the hint instruction upon a L2 replacement of the instruction line.

1   13. A branch prediction process for a computer system for improving branch prediction

2   rate when using a branch history table as defined by claim 4, further comprising

3   the generating operation being performed by the program instruction processor for

4   generating and executing hint instructions.

5   14. A branch prediction process for a computer system for improving branch prediction

6   rate when using a branch history table as defined by claim 4, further comprising

7   the generating operation of generating hint instructions being performed by a hint

8   processor operating in parallel with the program instruction processor.

1   **15. A branch prediction process for a computer system for improving branch prediction**

2   **rates when using a branch history table as defined by claim 14, further comprising**

3   executing a hint instruction when the hint instruction is received in the I-cache by testing

4   an operation code field in the hint instruction to determine if a completed hint instruction

5   is indicated or if a no-operation state is indicated for the hint instruction, and continuing

6   the executing process only if a completed hint instruction is indicated by performing the

7   following operations:

8   reading a BHT entry in the BHT located at an index determined by a bht_index field in the

9   hint instruction, and storing the BHT entry in a curr_bht register,

10   logically ANDing an Nth bit in an inversion of the branch_mask field in the hint instruction

11   with an Nth bit in the curr_bht register, where N is the bit position of the current

12   instruction in the instruction line, and logically ANDing the Nth bit in a branch_mask field

13   with an Nth bit in a bht_bits in the hint instruction,

14   logically ORing outputs of the two logical ANDing operations to provide an Nth bit output,

15   and setting an Nth bit in a new_bht register to the Nth bit output,

16   receiving without change in the new_bht register at bit locations other than at the Nth bit

17   location the bits in the curr_bht register at corresponding bit locations other than the Nth

18   location, and

1   setting the content of the new_bht register into the current BHT entry in the BHT to

2   restore the BHT entry to its last prediction state for the current instruction.

3   16. A branch prediction process for a computer system for improving branch prediction

4   rates when using a branch history table as defined by claim 15, further comprising

5   performing all of the hint instruction operations in a hint instruction processor.

6   17. A branch prediction process for a computer system for improving branch prediction

7   rates when using a branch history table as defined by claim 16, further comprising

8   performing all hint instruction operations and all program instruction processor

9   operations in a single semiconductor chip.